Evolution Towards Global Routing Scalability

Varun Khare, Dan Jen, Xin Zhao, Yaoqing Liu, Dan Massey, Lan Wang, Beichuan Zhang, Lixia Zhang

***Abstract*—Internet routing tables have been growing rapidly due to factors such as edge-site multihoming, traffic engineering, and disjoint address allocations. To address the routing scalability problems caused by this rapid growth, we propose an evolutionary approach that is incrementally deployable and provides immediate benefits to any adopting ASes. The basic premise of the approach is that route aggregation removes from routing tables the unnecessary topological details about remote portions of the Internet. We demonstrate that aggregation can be applied incrementally starting from local scopes within individual routers and individual ASes, and gradually expanded to the global Internet scope. The evaluation studies show that route aggregation is effective in addressing FIB scalability problems within a router and within a network.**

I. Introduction

THE INTERNET routing system is facing serious scalability problems as reported in the IAB Routing and Addressing Workshop [18]. For the past 15 years, the global routing table size in the Default Free Zone (DFZ) has been growing at greater than linear rates [13]. The main factors contributing to this rapid growth are the increasing number of organizations connected to the Internet and the increasing practices of multihoming and traffic engineering. An edge network with multiple external links to different providers is said to be multihomed [23]. Such a multihomed network may have one or more Provider-Independent (PI) address prefixes, and/or obtain one or multiple Provider-Assigned (PA) address prefixes.

In order to be reached through any of its multiple providers, the edge network injects its prefix(es), regardless of whether they are PI or PA prefixes, directly into the global routing system. Such operational practices destroy providers' attempt to aggregate prefixes based on topological connectivity. Furthermore, the edge network may split (*i.e.*, de-aggregate) one prefix into multiple and announce them separately through the providers to load-balance incoming traffic over its multiple provider links, causing the global routing table to grow faster than the number of connected organizations. Other factors, such as disjoint address blocks allocated to the same organization, also contribute to the routing table growth. The increasing routing table has also exposed the core of the network to increasingly frequent routing updates, many of which originated from a small number of highly unstable edge sites [20].

By injecting non-aggregatable prefixes into the global routing system, edge networks gain multihoming and traffic engineering benefits, but in general do not experience any negative impact from such actions. It is the ISPs, especially large ISPs, that bear the consequence of the rapid growing global routing tables. Some ISPs are already facing difficulties in handling the large routing and forwarding tables along with the associated routing churns. The increasing deployment of IPv6 will only exacerbate these problem, as it removes restrictions over IP address allocations. Therefore, ISPs need effective means to control their own operational costs in face of rapidly growing routing tables.

A scalable Internet routing system should not constrain the growth of the Internet. More specifically, each network should be able to control its own routing table size and the associated routing dynamics regardless of the growth of other networks and their traffic engineering practices. In this paper, we explore the use of prefix aggregation to achieve this goal. Aggregation allows the use of a shorter prefix to replace multiple longer prefixes in the routing and forwarding tables.

One intuitive solution is to separate the edge networks from the transit core in the inter-domain routing system [15]. This separation can remove edge prefixes (the main source of routing table growth and churn) from the core routing table. As such, the core routing table will become topologically aggregatable and it will not be affected by edge dynamics. Several previous and existing efforts, *e.g.*, APT [14] and LISP [11], seek to develop scalable routing solutions along this direction. They make use of the Map & Encap [8] approach, *i.e.*, mapping edge networks to their attachment points to the core and using packet encapsulation to deliver data from edge to edge across the core. One drawback of this class of solutions, however, is that it is effective only when the majority, if not all, of the networks have deployed it. Although controlling the routing table size is a commonly shared goal, especially in lieu of the increasing IPv6 deployment, different networks have different degrees of incentive as well as affordability in solution deployment, and some parties may not even see the need to take any action towards fixing the problem for the time being. Thus, for a solution to get deployed in the real world, it must be able to provide *immediate* benefits for the first-mover rather than waiting for other networks to deploy.

The realistic deployment of a scalable routing solution needs to be *evolutionary* in nature. By evolutionary we mean that (1) the solution should be deployable by an individual AS without needing coordination with any neighboring ASes; (2) even within a single AS, the solution should enable the routing table size reduction at only those routers whose capacity falls

Manuscript received ; revised .
Varun Khare is with (e-mail:vkhare@cs.arizona.edu).
Dan Jen is with (e-mail: jenster@cs.ucla.edu).
Xin Zhao is with (e-mail: zhaox@cs.arizona.edu).
Yaoqing Liu is with (e-mail: yliu6@memphis.edu).
Dan Massey is with (e-mail: massey@cs.colostate.edu).
Lan Wang is with (e-mail: lanwang@memphis.edu).
Beichuan Zhang is with (e-mail: bzhang@cs.arizona.edu).
Lixia Zhang is with (e-mail: lixia@cs.ucla.edu).
Digital Object Identifier 10.1109/JSAC.2010.1010xx.

0733-8716/10/$25.00 © 2010 IEEE

behind the FIB or RIB growth curve; (3) the solution should be an incremental step built on top of the existing system, so that it is cheaper and easier to roll out; (4) the AS adopting the solution can receive immediate benefits that are higher than or comparable to the cost of deploying the solution; and finally, (5) the solutions must work transparently with the rest of the system even when large parts of the system do not adopt the solutions at the same time. Towards this end, we present "Aggregation with Increasing Scopes (AIS)," an evolutionary path towards scaling the global routing system, where prefix aggregation is applied incrementally starting from local scopes within individual ASes and gradually expands to the global Internet scale.

One fundamental difference between our evolutionary design and the Map & Encap class of solutions is that the separation of edges from the core is not the starting point of our design; rather, it can be a consequence of expanding the prefix aggregation scope (see Section III-E). Unlike the traditional "incremental deployability" claimed by many new designs, which merely provide inter-operation between sites that have adopted the new design and legacy sites without addressing deployment incentives, an evolutionary path gradually progresses *towards* a new routing system structure with immediate incentives provided at each step, even though the full details of the end may not be known at this time.

In this paper, we sketch out an evolutionary path and demonstrate the feasibility of moving the routing system towards a scalable architecture through incremental steps, resolving the FIB scaling problems within a router through FIB Aggregation and within a network through Virtual Aggregation, and eventually resolving the RIB scaling problems. We articulate the reasons why this evolutionary process should not get stuck at a "local optimum", although we cannot offer a theoretical proof. Our evaluation shows that basic Level-1 FIB Aggregation can reduce routers FIB size by 30% to 50% while the Level-4 FIB Aggregation can reduce routers FIB size from 60% up to 90%, and Virtual Aggregation can reduce the FIB size on a network-wide basis by 80% for few specialized routers and up to 93% for all other routers.

The rest of the paper is organized as follows. Section II introduces the difficulties in deploying solutions in the Internet. Section III presents an evolutionary path to resolve the routing scalability problem by applying aggregation incrementally. Section IV evaluates the effectiveness of aggregation in reducing FIB size within a router and within an AS. We discuss related work in Section V and conclude in Section VI.

II. Challenges in Achieving Routing Scalability

Two fundamental properties of the Internet are its distributed governance and its diversity along multiple dimensions. These properties have led to different degrees of routing scalability problem and varying affordability of new solutions at individual networks. This heterogeneity suggests that the Internet routing infrastructure needs an *evolutionary* path to move forward. We also note that there is a fundamental difference between the evolutionary process and the conventional concept of "incremental deployability."

A. Not All Networks Are Equal

The Internet is an interconnection of tens of thousands of independently administered networks, each with its own budget, planning, business models and operational practices. Different networks may perceive the growing scalability problem differently. For instance, edge networks and small regional Internet service providers typically do not carry the full BGP routing table; instead they propagate only internal routes inside their networks and use default routing to reach the rest of the Internet through one or a few exit points. On the other hand, large networks in general carry full BGP routing table internally to efficiently forward data traffic to the large number of exit points and to propagate routes to neighboring networks. As a result, the former may not care about the size of global routing tables but the latter may feel the pain from its growth. Among the networks that do carry the full routing tables internally, some (e.g. content providers) are able to upgrade their routing infrastructure to keep up with the growing demands of the BGP tables, while others may not be able to afford doing so. These observed differences are supported by the results from a survey we conducted in early 2009 on routing scalability among a small group of people with operational expertise [27].

Even among networks that face the routing scalability problem, there can still be different severity at different routers. For example, we learned from a few large ISPs that, although they were able to upgrade the relatively small number of core routers with the latest technology that can handle a million or more routes, they were unable to upgrade all their edge routers that may count up to a thousand or more; some of these edge routers are more than 10 years old. Furthermore, even if a network may suffer pain from the growing routing table size, it still may not be able to deploy a new solution if the cost is considered prohibitively high; there is no direct correlation between the routing table size growth and revenue growth. Given the scale and diversity of the Internet, it is certain that the buy-in of any new solution will not be harmonious. Even for those networks that require a solution to handle their routing scalability, the deployment will likely be a gradual process consisting of several stages.

B. Evolutionary Design $\neq$ Incremental Deployability

It is important to distinguish between an "evolutionary process" towards a final solution and a "revolutionary new design" with incremental deployability. One fundamental difference is that all brand new designs tend to have an implicit assumption that the entire system would eventually move to the new design. Therefore, it is likely the case that the assumed benefit of the new design would be fully achieved only after a majority, if not all, of the system has deployed the design, and only then the cost of the deployment would be amortized. The incremental deployment machinery simply glues together the part that has made the change and the rest that has not, so that the system can function together at the intermediate, and hopefully transient, stage. However, the system as a whole would be in a sub-optimal state until the new design gets fully deployed.

In contrast, an evolutionary approach recognizes that changes to the Internet can only be a gradual process; at each step the networks making the changes not only need to be able to inter-operate with the rest of the world, but also need to get immediate benefits to justify the deployment cost. Such benefits from making the changes cannot be in contingency on the future behaviors of many other networks. A revolutionary new design would provide a clear and clean picture of a new routing system if and once the final stage is reached. On the other hand, an evolutionary process presents a much messier and more complex picture, both because old protocols are twisted for new functions and because different networks may be at different stages of the evolution. Although the exact final picture of the evolutionary process is not known a priori, its overall trend is to move towards a system where an increasing number of places perform aggregation of various kinds. Even though individual networks adopt aggregation for self-interest, as more and more networks make the change, the global routing system will most likely move towards a scalable structure.

III. An Evolutionary Path to Routing Scalability

Prefix aggregation offers a means to abstract out the unnecessary topological details about remote portions of the network thereby substantially reducing the routing table size. Aggregation allows a shorter prefix to cover up a number of longer prefixes in a routing table. However, the operational practices of today's edge sites prevent service providers from applying topological aggregations of edge networks' prefixes into their own address prefixes. In this section we sketch out an evolutionary path towards scaling inter-domain routing where aggregation is applied incrementally starting from local scope within an AS and gradually expanding to the global Internet scope in an evolutionary fashion to address the FIB, RIB and churn scaling problems. Within a given scope of deployment, specific longer prefixes can be eliminated and aggregated under a shorter and less specific prefix as long as it does not affect the delivery of the data traffic.

At this time, we can see several steps in evolving today's BGP routing system towards a controllable growth of the routing table size. We identify potentially most severe pain at each step that warrants a fix. We then identify a fix that has a reasonable cost, can be carried out by individual networks, and can be built on top of the existing operations, so that it does not break any other parts of the global routing system. Note that any such simple fix necessarily has its limitations. As the fix gets widely deployed, its limitations are likely to become more pronounced, and can become the next problem to address. At the same time, other aspects of the routing scalability problems that were not addressed by these fixes may become more severe. These issues will lead to the next step of evolving the system forward.

A. Router FIB Aggregation

Forwarding tables are derived from routing tables and router configurations, so their sizes increase as routing tables grow. But forwarding tables use high performance memory that is more expensive and more difficult to scale than the memory used to hold routing tables. As a result, ISPs are forced to upgrade their routers at a faster pace, which increases their operational costs. Since the growing FIB size is of utmost concern for ISPs, we consider it the first issue to be addressed.

A purely local solution is FIB aggregation ([9], [28]), which combines multiple entries in the routing table to one entry in the forwarding table without changing the next hop for data traffic. Intuitively, a FIB aggregation scheme works as follows: if all longer prefixes, say under 1/8, share the same next hop with the covering prefix 1/8, then only 1/8 needs to be installed in the FIB and all the longer prefixes under 1/8 can be removed from the FIB.

The effectiveness of FIB aggregation depends upon two factors: (1) what prefixes are present in the routing table, and (2) how these prefixes are distributed over the next-hop routers. In general, the fewer neighbors a router has, the better FIB aggregation it may achieve. In the extreme case, if all prefixes share the same next-hop, then the degree of FIB aggregation is maximized. According to Li *et al.* [16], although some routers have high degrees up to a few hundred, these routers connect to a large number of end-customers, not transit neighbor routers. Therefore, they will still use only a small number of next-hops, i.e., the transit neighbors, to reach most of the address prefixes.

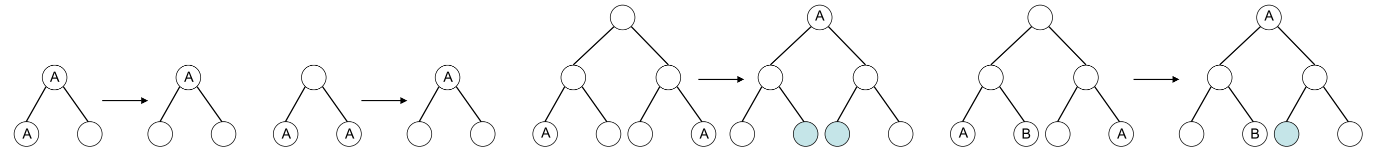
Besides sharing the same next-hop, prefixes also need to be *numerically* aggregatable (unless we can include the gaps between them in aggregation). This is possible due to two factors. First, in IP address allocation, large blocks of Internet addresses are first allocated to RIRs and then they further allocate the addresses to networks within the same region. A router outside the region tends to use the same next-hop to reach these address prefixes, which can then be aggregated. Second, for prefixes split for traffic engineering or other purposes, a router near the origin network is likely to have different next-hops, but a router further away from the origin network is more likely to have the same next-hop towards these numerically aggregatable prefixes. Therefore, although FIB aggregation is opportunistic and the aggregation degree varies from router to router, there are some inherent properties of the Internet that can make FIB aggregation effective.

FIB aggregation should ensure packet delivery and not change the paths that packets take, which we call **forwarding correctness**. We define two types of forwarding correctness as follows.

- *Strong Forwarding Correctness*: The longest-prefix lookup of any destination address that appears in the original FIB should return the same next-hop before and after the aggregation. Moreover, any destination address that does not appear in the original FIB should not appear in the aggregated FIB.
- *Weak Forwarding Correctness*: For destination addresses that appear in the original FIB, the longest-prefix lookup should return the same next-hop after the aggregation.

We design and implement four algorithms at different FIB aggregation levels as shown in Figure 1. The first two satisfy strong forwarding correctness, while the last two satisfy weak forwarding correctness.

- Level-1 aggregation removes prefix p if it shares the same



(a) Level-1: Removing covered prefixes (b) Level-2: Combining sibling prefixes (c) Level-3: Allowing extra routable space (d) Level-4: Allowing holes in the aggregation

Fig. 1. Different Levels of FIB Aggregation. The binary tree represents part of the IP address space. Nodes labeled with letters are prefixes in the routing table, and the letter represents the next-hop for the prefix. Nodes without labels do not have their corresponding prefixes in the routing table. Filled nodes are extra routable space introduced by the aggregation.

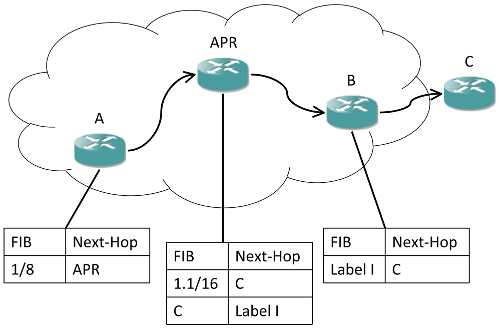
next-hop with its immediate covering prefix $p\prime$, which is the longest prefix that is less specific than the prefix p.

- Level-2 aggregation combines sibling prefixes that share the same next-hop into a parent prefix if the parent prefix is nonexistent in the routing table. Sibling prefixes are of the same length, numerically consecutive and numerically aggregatable.
- Level-3 aggregation combines a set of non-sibling prefixes that share the same next-hop into a super prefix if the super prefix is nonexistent in the routing table. Level-3 aggregation introduces non-routable space between the non-sibling prefixes.
- Level-4 aggregation combines a set of non-sibling prefixes with the same next hops into a super prefix even if other prefixes in-between exist with different next-hops. Level-4 aggregation may also introduce extra non-routable space underneath the super-prefix.

The difference between weak and strong forwarding correctness is that the former (*e.g.*, Level 3 and 4 aggregations) introduces new prefixes that cover previously non-routable space, therefore some previously non-routable traffic will be forwarded. On the other hand, allowing extra routable space improves aggregation. For example, Draves *et al.* [9] have designed an algorithm, called ORTC, that aggregates FIB to the furthest extent under *strong* forwarding correctness. But our Level-4 aggregation can aggregate the FIB more than ORTC.

The impact of extra routable space depends on how much traffic is destined to that address space. In normal operational conditions, the volume of such traffic should be negligible. However, malicious traffic such as port scanning can have non-routable destinations and in certain cases it may become noticeable. Eventually these packets will be dropped, either because they arrive at a router that does not have a route for these packets, or because the packets' time-to-live expires, but they will consume bandwidth during transit. One of our ongoing research efforts is to evaluate the potential impact of the extra routable space. Note that it is possible to limit the size of extra routable space. For example, one can stop aggregation for prefixes whose lengths are shorter than a threshold. We found that the best tradeoff between table size reduction and extra routable space size is achieved when the aggregation stops at the prefix length of 15. Furthermore, null-routed prefixes can be inserted to remove the extra routable space.

As Internet routes change over time, there is a need to update the aggregated FIB to handle the changes in the routing tables. Re-running the full FIB aggregation results in the best aggregation but at the expense of significant computation overhead. We present network operators with the following options to control the cost of updating the aggregated FIB. First, operators can choose the level of FIB aggregation needed at specific routers, e.g. routers with slower CPUs and more routing updates can use lower levels of FIB aggregation, which are generally faster than higher level algorithms. Second, we have designed algorithms to incrementally update the aggregated FIB upon receiving a routing update. Third, the full FIB aggregation is invoked only when needed, e.g., when the table size reaches a threshold after being incrementally updated for a while or when router load is under a threshold. The details of incrementally updating the aggregated FIB is available in [28]. We note that previous FIB aggregation efforts, *e.g.*ORTC [9], did not take incremental update into consideration when designing their algorithms.



FIB	Next-Hop
1/8	APR

FIB	Next-Hop
1.1/16	C
C	Label l

FIB	Next-Hop
Label l	C

Fig. 2. Virtual Aggregation

FIB aggregation is limited to individual routers. Therefore, it requires no changes to routing protocols or router hardware, nor does it impact multihoming, traffic engineering, or any other network-wide operational practices. It can be done by a software upgrade at those routers facing the FIB scalability issues, and it is compatible with any future solutions that may change routing. Deployment of FIB aggregation is evolutionary since it provides immediate benefits to network operators who are trying to meet the FIB storage requirements on the old routers within their networks.

B. Network Coordinated FIB Aggregation

The effectiveness of FIB aggregation depends upon the aggregatablity of covered and covering prefixes in the routing table, and therefore has a lower bound on how much it can reduce the FIB size. Network operators wanting to further reduce FIB size can adopt Virtual Aggregation proposed by Ballani *et al.* ([4], [5]).

TABLE I
VIRTUAL PREFIX COVERAGE

Virtual Prefix	No. of Covered Prefixes	% of Global Routing Table (GRT)
0/4	30K	11.1%
64/4	80K	29.6%
128/4	30K	11.1%
192/4	130K	48.2%

Virtual Aggregation works as follows (see Figure 2). An ISP can reduce its routers' FIB size by configuring a router, dubbed as Aggregation Point Router (APR), to announce a short prefix, say 1/8, into its own network in place of multiple longer prefixes that fall within 1/8. This short prefix is called a virtual prefix. The APR installs FIB entries for all the longer prefixes (e.g. 1.1/16) covered by the virtual prefix it announces. And the non-APR routers only maintain the route for the virtual prefix and suppress the routes for longer prefixes, covered by the virtual prefix, in their FIBs. Note that these suppressed routes are still in their RIBs; they are not installed into their FIBs. When a router A receives a packet to be forwarded to 1.1/16, A's FIB directs the packet to the APR, and the APR then encapsulates the packet to the egress router B for delivery to the external router C for the actual prefix 1.1/16.

In Virtual Aggregation, the choice of virtual prefixes impacts the FIB storage requirements of every router within the network. The non-APR routers only store routes to announced virtual prefixes and the APR routers only store routes to longer prefixes that fall within their announced virtual prefix. Virtual Aggregation provides the ISP a tuning knob in the form of choice of virtual prefixes as a means to control FIB storage requirements within its network. For instance, an ISP can choose to distribute its FIB storage requirements into 4 parts where the entire address space is divided into 4 virtual prefixes (VP) 0/4, 64/4, 128/4 and 192/4, or it can divide the address space into 256 /8s and announce 256 virtual prefixes. Table I presents the percentage of global routing table (from RouteViews [3] on Dec. 1, 2008) that each one of those 4 VPs cover. Similar distribution of global routing table amongst the virtual prefixes is seen from data on June 20, 2008 and Feb. 29, 2009. For even distribution of FIB storage requirements amongst APRs, each one of the 4 APRs would store about 25% of the global routing table. Any assignment of the 4 VPs to APRs does not evenly divide the FIB storage requirements since the APR assigned 0/4 stores only 11% of the global routing table, while the APR assigned 192/4 stores 51.3% of the global routing table by itself. However, dividing up the address space into smaller pieces by means of having longer virtual prefixes allows more fine-grained assignment of FIB storage requirements to each APR (see Section IV-B). So the former choice requires non-APR routers to store fewer virtual prefixes, but makes it harder to evenly distribute FIB storage requirements amongst APRs. And the latter choice allows FIB storage to be distributed more evenly amongst 256 APR routers, but more virtual prefixes need to be announced and stored in all the non-APR routers.

Virtual Aggregation is not completely without its drawbacks. Virtual Aggregation "stretches" packet delivery by sending packets via sub-optimal paths since all the packets destined to the prefixes that have been aggregated will go through the APRs. For instance, the packet from router A in Figure 2 destined for 1.1/16 traverses the sub-optimal path A–APR–B rather than the native path A–B. Furthermore, an APR becomes a single point of failure under Virtual Aggregation for the delivery of any packet destined to prefixes covered by the APR's announced virtual prefix. Virtual Aggregation again provides the ISP with a tuning knob in the form of APR placement choice to reduce the amount of added packet stretch and to provide robustness against APR failures. Packet stretch depends upon APR location, *e.g.*, if the APR is present on the native path of the packet, then the packet experiences no stretch. To increase the likelihood of APR being present on the native path of packets, APRs can be placed closer to ingress or egress edge routers. However, placing APRs close to every edge router in the network eats into the savings of ISPs and can also complicate the network. Virtual Aggregation also allows ISP to configure a smaller fraction of popular prefixes to be installed in the FIB of every router thereby allowing the bulk of the traffic to be routed natively without any stretch. Virtual Aggregation provides robustness against APR failures by allowing multiple redundant APRs to announce virtual prefixes within the network. Deploying multiple APR sets not only makes the network more robust but also improves the possibility of APR being present on the native path of packets thereby impacting the stretch-savings tradeoff in the network. An ISP must consider these factors when deciding how many APRs to deploy and where to place them within its network.

Virtual Aggregation increases the scope of aggregation to within a network domain, i.e., the virtual prefix that allows the longer prefixes to be aggregated out of the FIB is used by all the non APR routers within the network. Deployment of Virtual Aggregation is again evolutionary since it can be adopted within a single ISP without any dependency upon outside parties to deploy anything or act in any way. Aside from independent deployability, ISPs that adopt Virtual Aggregation do not need to wait for widespread deployment. The ISP adopting the solution immediately receives the benefits of FIB scalability within its own network. Both FIB Aggregation and Virtual Aggregation can be seen as complementary solutions and ISPs can choose which solution to deploy in parallel or sequentially for scaling its FIB storage requirements. FIB aggregation can be performed through a software upgrade at routers while Virtual aggregation requires network-wide router configurations and specialized routers to announce virtual prefixes.

C. Local RIB Aggregation

After a network X has deployed virtual aggregation for a while and has gained sufficient operational experience, it may become clear that many of its routers no longer need to maintain the full RIB table. For instance, if an internal router has a small FIB and relies on APRs to route packets towards all other destinations, it does not need a full RIB to build its FIB. Theoretically speaking, all border routers of network X that connect to legacy networks (i.e. those that have not

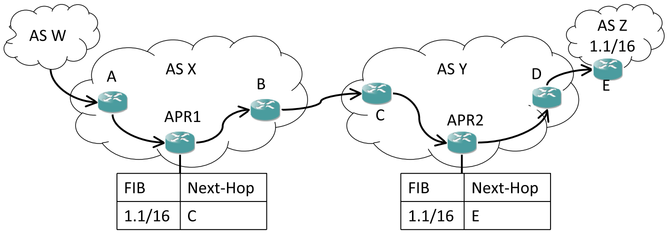


Fig. 3. Path Stretch and Encap/Decap Cost overhead with Virtual Aggregation

deployed Virtual Aggregation) would still need to keep the full RIB in order to make BGP announcements into the legacy neighbors. However, in practice, only those customer-facing border routers need a full RIB where the customers want to receive the full RIB. In case the customer uses network X as the default route for the rest of the Internet, then the customer-facing border router of network X only needs to store the virtual prefixes announced within network X, network X's internal routes and the customer's prefix in its FIB. Other border routers that face either peer or provider legacy neighbors only need to announce X's own customer prefixes to the respective neighbors. Careful engineering analysis and configuration can eliminate the need for many routers to keep the full RIB. And only the routers serving as APRs within the network X need to maintain the full RIB.

D. Reducing Virtual Aggregation Overhead

By application of FIB Aggregation and Virtual Aggregation we are able to reduce the FIB size on all the routers within an AS and limit full RIB size storage on only APR routers and those customer-facing border routers where customers want the full RIB. However, when two or more adjacent ASes all deploy Virtual Aggregation, packets that traverse these ASes will experience the cumulated path stretch and encapsulation/decapsulation cost of all the ASes along their paths. For instance, consider the packet destined for 1.1/16 and traversing the AS path W-X-Y-Z, as shown in Figure 3, where adjacent ASes X and Y have deployed Virtual Aggregation. The packet experiences a stretch of A–APR1–B in AS X and C–APR2–D in AS Y; faces encap and decap cost at APR1 and egress router B in AS X and APR2 and egress router D in AS Y respectively. The need to resolve this new problem of cumulated path stretch and encap/decap cost overhead can naturally lead to the next step of evolution towards better routing scalability.

Virtual Aggregation is limited to within a single network and therefore each packet is encapsulated to a local egress router within the network. And so within each AS where virtual aggregation is deployed, the packet has to go through local APR and thereby face stretch and encap/decap cost. In order to minimize this cumulative path stretch and encap/decap cost, the packet should be encapsulated directly to the egress router of the provider network serving the destination site. For instance in Figure 3 rather than having the packet encapsulated to X's egress router B, if X's APR1 encapsulates the packet directly to the egress router D of Y that connects Y to the destination site Z, then the path stretch is reduced and the packet will need to be encapsulated/decapsulated only once instead of two times. To enable such inter-AS Virtual Aggregation, X's APR needs to know Y's egress router for 1.1/16, and therefore mapping information that maps a destination prefix to its provider's egress router needs to be propagated across networks. One approach, as suggested in [26], is to piggyback such mapping information on existing BGP announcement for the prefix in the form of a Tunnel Endpoint Attribute that carries the address of the tunnel endpoint for the prefix and is transitive across ASes.

We reason the feasibility of this step as follows. First, this step towards better routing scalability will take place only after at least two adjacent networks (X and Y in our example) have deployed VA and benefited from it. Therefore, it is highly likely that they would not want to move away from VA but would like to minimize VA's cost in path stretch and encapsulation, in order to improve network performance for their customers. Second, the required BGP implementation changes are backward compatible, meaning that networks that have deployed this solution can easily interwork with networks that have not deployed this solution. Furthermore, adjacent VA enabled ASes may not need to exchange mapping information for all their prefixes. For an instance, the mapping information may be exchanged for only popular prefixes to improve the performance of the bulk of the traffic.

E. Inter-AS RIB Aggregation

Piggybacking the virtual aggregation mapping information on BGP can work well when the mapping table is small. When more networks have adopted Virtual Aggregation, the mapping table is likely to grow large, which may make it no longer feasible to piggyback all the mapping information on the existing BGP sessions. The main problem, as we can perceive today, would be the RIB size growth: a BGP router will receive the same mapping information from multiple neighboring BGP routers, and store all of it in its Adj-RIBs-IN. Thus BGP routers may end up with storing multiple copies of the same mapping information. For example, assuming ASes W, X, Y, and Z have a full-mesh connectivity among themselves, and AS W propagates a mapping entry [egress router R, customer prefix P], then X will receive 3 copies of this mapping entry from Y, Z, and W, respectively.

The natural next step is to move the mapping dissemination from the regular BGP instance (which is used for inter-domain routing) to a separate BGP instance only between APRs via multi-hop BGP sessions. Though the protocol can still be BGP for ease of deployment, APRs would run a different session (e.g., on a different TCP port) for mapping dissemination purposes only. Other regular routers run regular BGP instances for inter-domain routing purpose, but are relieved from bearing the overhead of storing and propagating mapping information or the full RIB table.

When the RIB size for most routers (other than the APRs) is reduced, what are the prefixes that get dropped out of the RIB? Since APRs (or ingress routers) must encapsulate packets towards egress routers that connect to the more specific prefixes that have been aggregated out, the ASes must exchange the reachability information about their own egress routers, so that routers in different ASes know how to reach each other. The

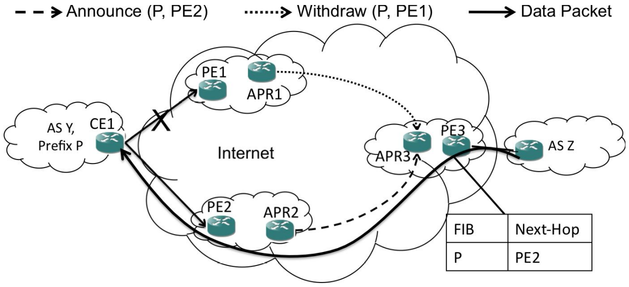


Fig. 4. Pushing Mapping Updates due to Edge Dynamics

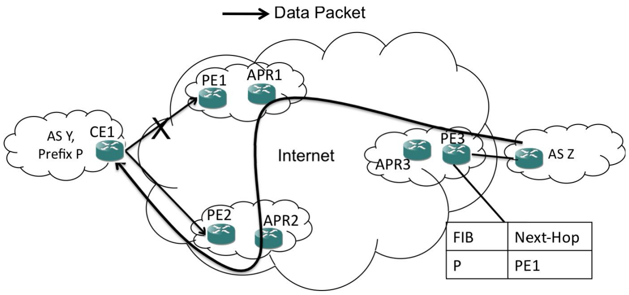


Fig. 5. Suppressing Mapping Updates due to Edge Dynamics

prefixes that got aggregated out of the core routing system would be those that belong to the edge ASes. As such, Virtual Aggregation plus mapping exchanges effectively drives the overall routing system towards the separation of edge site prefixes from the transit network routing, a scalable routing architecture that the APT design has depicted [14].

F. Insulating the Core from Edge Churn

In the current Internet, flaps of customer prefixes are propagated to the rest of the Internet in the form of BGP updates, *i.e.*, routing churn. With virtual aggregation and mapping exchange, the churn would be reflected as mapping updates, which are disseminated through the interconnections of APRs. We perceive this as a benefit, as other non-APR routers can be sheltered from updates due to edge instabilities. Our earlier measurement and analysis study [21] has shown that most Internet topology growth comes from the addition of customer edge ASes. It is conceivable that as the number of customer sites continue to increase, the amount of churn may become too much to handle in a cost-effective way. A solution to this edge churn problem is to insulate the mapping dissemination system from the edge dynamics. Based on the current BGP data, our estimation shows that, if we could remove BGP updates induced by customer prefix instabilities, we would have reduced the total amount of routing churn by an order of magnitude [17].

Ideally, when the link connecting a customer site to a provider fails, the mapping system should propagate this failure information only when the failure has a long duration, so that every network will be aware of this failure and choose an alternative path to reach the affected customer site. For instance, in Figure 4 when the link CE1-PE1 fails then APR1 withdraws the mapping (P, PE1) causing ingress router PE3 to choose the alternative mapping (P, PE2) and send traffic through PE2 to reach customer site AS Y. In contrast, short failures, which are frequent, should not be propagated but rather suppressed within the mapping system. For instance, in Figure 5 when the link CE1-PE1 fails then APR1 rather than withdrawing the mapping (P, PE1) attempts to find an alternate APR for the prefix P which is APR2 and tunnel the packet to that APR.

TABLE II
CHARACTERIZING DAILY PREFIX TRAFFIC

	% of Traffic
Top 500 prefixes destinations	63.3%
Top 4K prefixes destinations	88.5%
Remaining 151K prefixes	11.47%

Determining the duration of link failures is hard, therefore we propose other means to handle such link failures, *e.g.*, data-driven failure handling reports a link failure to an edge network only when there are data packets heading towards the failed link. Rexford *et al.* [22] have reported that a small number of popular destinations responsible for bulk of the Internet traffic have remarkably stable BGP routes and the vast majority of BGP instability stems from a small number of unpopular destinations. Table II presents the break down of the Internet traffic on June 18, 2009 collected over the link between a regional Internet aggregation point and a Tier-1 ISP. Upon ordering the prefix destinations in terms of traffic received, we find that the Top 4K prefix destinations attract 88.5% of the Internet traffic while the remaining 151K prefix destinations attract only 11.47% of the Internet traffic. The Top 4K prefixes are popular by virtue of attracting the bulk of the Internet traffic and their corresponding BGP routes are expected to be stable implying that most of the traffic to these popular destination will not trigger pushing of mapping updates. The remaining prefixes responsible for majority of the BGP instability attract only 11% of traffic which may potentially trigger mapping updates in the event of route failure. More information about data-driven failure handling mechanism can be found in our earlier paper [14].

G. Summary

The evolutionary path outlined above uses aggregation as the main technique to shrink routing and forwarding tables, and deploys different mechanisms that realize aggregation with increasing scopes. With FIB aggregation, the scope of aggregation is limited to within a router. With Virtual aggregation, the lookup of a longer prefix covered by virtual prefixes underneath the APR would result in an egress router within the AS. The scope of aggregation is within an AS, *i.e.*, the prefix lookup at the APR results in an egress router of the AS. However, upon exchanging the mapping information of destination prefixes to their corresponding provider egress routers, the scope of aggregation is expanded to cover adjacent ASes participating in such exchange. Now the lookup of the same prefix underneath the APR results in the egress router of the provider connected to the edge site. Therefore, as more and more adjacent ASes start exchanging mapping information, the scope of aggregation will increase and gradually encompass the core networks.

IV. Evaluation

In an evolutionary approach, the new concepts and solutions are introduced in stages. As a result, realistic evaluation of solutions proposed for later stages can be done only after the solutions proposed for earlier stages have been deployed. At present, we can realistically investigate only the proposed solution to the FIB scalability issue since it can be directly applied to the current networks. In this section, we first evaluate the effectiveness of applying aggregation locally at a router to resolve its FIB scalability problem. We use publicly available routing tables from tens of networks to evaluate the various FIB aggregation algorithms for their table size reduction and computation times. We also use BGP routing updates to evaluate the FIB update algorithm. Thereafter, we evaluate the effectiveness of applying aggregation within an AS to resolve a network-wide FIB scalability problem. We evaluate the FIB savings and the added stretch that a real Tier-1 ISP would experience.

A. *FIB Aggregation Evaluation*

In our evaluation, publicly available BGP routing tables are taken from the route-views.oregon-ix.net collector of the RouteViews project [3]. These routing tables contain valid next-hop ASes but they do not have next-hop router information and they also may not reflect the diversity of next-hops that an operational router typically would have, since the monitored routers may not be operational routers. Therefore, we need to generate realistic next-hops based on known information. Our guideline for generating realistic next-hops is to overestimate the number of next-hops so that the table reduction results reflect the worst case scenario, and real routers are likely to have better aggregation ratio.

In order to utilize the RouteViews routing tables in our evaluation, we make an assumption that the prefixes sharing the same next-hop AS are likely to share the same iBGP neighbor and thus will share the same next-hop router. We validate this assumption using routing tables downloaded from route servers [1], which contain the iBGP neighbor address for each prefix. Assuming intra-domain routing uses a single best path, prefixes that share the same iBGP neighbor will share the same next-hop router. Now we need to find out whether prefixes sharing the same next-hop AS also share the same iBGP neighbor. For each next-AS hop, if there is only one iBGP neighbor, then all the prefixes using this next-AS hop share the iBGP neighbor. In case there are multiple iBGP neighbors connected to the same next-AS hop then the one that carries the most prefixes is called “popular”, and we expect that most of the prefixes use the popular iBGP neighbors. We found that more than 90% of the prefixes indeed use the most popular iBGP neighbor in all the default-free route server tables. Therefore, for evaluation purposes, we use the next-AS hop for each prefix as the next-hop router. However, it is worth noting that approximating next-hop router using next-AS hop tends to underestimate the effectiveness of aggregation schemes, since large networks have hundreds to thousands of neighbor ASes, but the number of real next-hops should be much smaller.

Once we have identified next-hop routers and applied the FIB aggregation strategies, we verify the correctness of each aggregated FIB by dividing each original RIB prefix into multiple /24 sub prefixes and look up the /24 sub prefixes in the aggregated FIB. The next-hop in the aggregated FIB should give the same next-hop as that in the RIB. All the results from our FIB aggregation algorithms and incremental update algorithms have been verified using this method. The evaluation has been done on a Linux machine with an Intel Core 2 Quad 2.83 GHz CPU. The implementation uses a single thread that is bound to a single core at runtime. The algorithms have been implemented in C with no supplementary performance optimization techniques.

1) FIB Table Size reduction and Overhead: We applied the four levels of aggregation algorithms to 36 routing tables archived at RouteViews on Dec. 31, 2008 and calculated the ratio between the aggregated FIB size and the original routing table size. We obtained the following results (the figure is not shown due to space constraint): (1) each level of aggregation reduces the FIB size more than the previous level; (2) even with the simple Level 1 aggregation, the FIB size is reduced by 30% to 50%; (3) Level 4 aggregation reduces the FIB size by 60% to over 90% with a median around 70%. For some routing tables, almost all the prefixes share the same next-hop, so they can be aggregated into a few entries.

To evaluate the effectiveness of our algorithms over a longer period of time, we applied them to the RouteViews routing tables from 2001 to 2008. For each year, we used all the tables available on Dec. 31, and report the median aggregation ratio of all the routing tables in Table III. The result shows an overall slightly decreasing trend, suggesting that the FIB has become more amenable to aggregation over the years. One possible explanation is that the increasing practice of prefix splitting due to multihoming and traffic engineering has made a larger percentage of FIB entries aggregatable.

Table IV presents the measured computation time incurred by the aggregation algorithms to aggregate each of the 36 routing tables. The Level 1, 2 and 3 aggregation algorithms typically require tens of milliseconds while Level 4 aggregation algorithm consume 110 milliseconds on average. An operational router may have slower CPU than our commodity Linux machine, but it has specialized hardware and software, thus it is hard to infer a router’s computing time from what we report. Nevertheless, the simplicity of the algorithms and the very short computing time suggest that the computational overhead in an operational router may be small. Moreover, our results can be used to compare the relative speed between different aggregation algorithms.

2) Routing Update Handling: As BGP updates arrive, the RIB may change and thus the aggregated FIB may also need to change. In order to evaluate the incremental update algorithm, we used one month (December 2008) of BGP routing updates collected by RouteViews from a peer router at a large ISP (Level-3 Communications). There were a total of 7,254,478 routing updates during the month and there were no BGP session resets or table transfers during the month. We make the following observations. The RIB processing time per routing update is increased from 0.6μs without aggregation to 0.62μs for Level 1 aggregation (3.3% increase) and 0.64μs for Level

TABLE III
MEDIANS OF AGGREGATION RATIO FROM 2001 TO 2008

Algorithms	2001	2002	2003	2004	2005	2006	2007	2008
Level-1	0.686	0.707	0.686	0.653	0.672	0.660	0.669	0.665
Level-2	0.556	0.564	0.527	0.491	0.509	0.491	0.494	0.479
Level-3	0.498	0.518	0.470	0.433	0.462	0.447	0.455	0.437
Level-4	0.396	0.430	0.367	0.347	0.367	0.353	0.358	0.343

TABLE IV
COMPUTATION TIME

Algorithms	Max (ms)	Min (ms)	Median (ms)
Level-1	58.0	53.0	55.8
Level-2	74.0	63.0	66.0
Level-3	76.7	66.9	69.8
Level-4	120.0	110.0	115.5

TABLE V
PROCESSING ROUTING UPDATES IN DECEMBER 2008

Algorithms	Total RIB Proc. Time(s)	Avg. RIB Proc. Time (μs)	Total FIB Updates	Total FIB Proc. Time(s)	Avg FIB Proc. Time(μs)	Total Affected Prefixes in FIB
Original FIB	4.37	0.60	2914020	2.58	0.89	2914020
Level-1	4.47	0.62	2904632	2.45	0.84	2921339
Level-2	4.51	0.62	2901197	2.44	0.84	2933968
Level-3	4.64	0.64	2900302	2.42	0.83	2940223
Level-4	4.67	0.64	2897384	2.40	0.82	2941992

4 aggregation (6.7% increase). The increase is due to the need to update more than one node in the RIB tree, but the small increase suggests that the extra overhead for updating the RIB is minimal. The total FIB processing time (5th column) decreases by 5% (Level-1) to 7% (Level-4), despite a slight increase in the total number of affected prefix nodes (7th column). Although the total number of affected prefixes for the aggregated FIBs is slightly greater than the unaggregated FIB but for each prefix it takes less time to update in the aggregated FIB as per 6th column, which leads to a lower total FIB processing time. The lower FIB update time per prefix is likely due to the smaller FIB size after aggregation, which translates into faster prefix lookup. In summary, FIB aggregation can reduce both the FIB size and FIB update time with minimal extra RIB processing time.

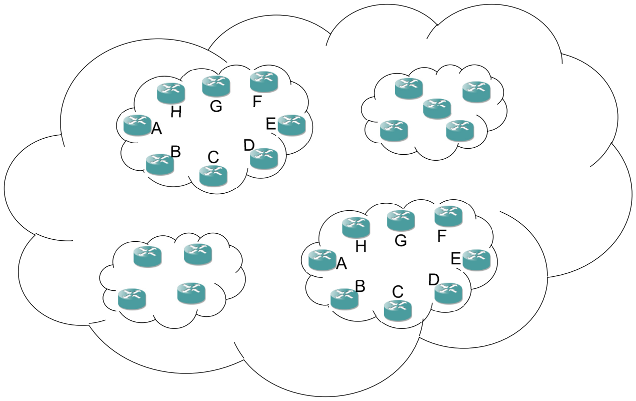


Fig. 6. Conceptual topology for VA evaluation. Full set of APRs deployed at major PoPs while no APRs deployed at non-major PoPs.

B. Virtual Aggregation Evaluation

We evaluate the FIB savings and added stretch experienced by a packet within a particular deployment scenario of Virtual Aggregation (VA) using the North America topology and routing updates of a major Tier-1 ISP. Prior evaluation of Virtual Aggregation, conducted by Ballani *et al.* [5], analyzes deployment of VA within another large Tier-1 ISP with the focus on determining optimal placement of APRs for an optimal set of virtual prefixes (VP). In [5], the selection algorithm for APR placement allows any router to be an APR for any given virtual prefix. The resulting assignment of APRs makes the network rather complicated to troubleshoot, as each router must anycast to its nearest APR for any given virtual prefix, and APRs are scattered throughout the network for virtual prefixes of different lengths. In contrast, we deploy APRs depending upon the network topology of the large Tier-1 ISP to reduce the complexity introduced by VA and choose virtual prefixes to evenly distribute the FIB storage requirements amongst the APRs. In [5] the stretch experienced by a packet is calculated using geographic distances rather than actual IGP delivery times, which are expected to be longer due to processing time on routers as well as longer paths. We calculate stretch using traceroute, which is much more accurate than the geographic distance measure.

1) Choosing a Realistic VA Deployment: We analyze the given Tier-1 ISP's topology to figure out a realistic deployment of APRs and choice for virtual prefixes to be announced within the network. We find that 85% of PoPs contain less

TABLE VI
DISTRIBUTION OF FIB STORAGE REQUIREMENTS AMONG APRS

APR	VPs Announced	# of Covered Prefixes
APR A	0/8-64/8	34321
APR B	65/8-74/8:	35840
APR C	75/8-119/8:	34410
APR D	120/8-189/8:	34836
APR E	190/8-199/8:	36999
APR F	200/8-203/8:	34405
APR G	204/8-210/8:	36069
APR H	211/8-255/8:	29520

than 5 routers with full routing tables, which we refer to as *non-major PoPs*, and the remaining 15% of PoPs were all located in heavily populated cities, such as Los Angeles, New York, and Chicago which we refer to as *major PoPs*. Due to the discrepancy between major PoPs and non-major PoPs, we make the following VA deployment decisions. For every virtual prefix, there needs to be an APR announcing it in each major PoP and only the major PoPs contain APRs. Each major PoP contains 8 different APRs since the smallest major PoP consists of 8 routers each of which can be used as an independent APR. This choice allows every major PoP to contain a full set of different APRs. Since there are only a handful of major PoPs in the Tier-1 ISP, the location and replication of APRs is limited to a well-known number and there is enough replication of APRs per virtual prefix to provide adequate robustness against APR failures. Figure 6 presents the VA architecture conceptually for Tier-1 ISP under the specific VA deployment scenario where the number of major and non-major PoPs shown do not represent the actual number. Routers in the major PoPs use local APRs while routers in non-major PoPs use APRs from their nearest major PoP for encapsulating traffic under the virtual prefix to the appropriate egress router. These VA deployment decisions simplify troubleshooting efforts for network operators. In our proposed VA deployment, correct packet delivery can be easily determined by knowing the ingress PoP of a packet whereas in [5] a router-by-router trace of the packet needs to be done to determine the correct path a packet should have taken.

Thereafter we attempt to evenly distribute the FIB storage requirements amongst 8 different APRs, in every major PoP, through the choice of virtual prefixes to be assigned to the deployed APRs. We split the IPv4 address space into 256 /8 virtual prefixes and assign them to the 8 APRs. The address space is divided through /8 virtual prefixes since it is the longest virtual prefix that is still as short or shorter than any real prefix in the global routing table. The uniform length also allows virtual prefixes to be identified easily. With more virtual prefixes, each APR is assigned a finer granularity of virtual prefix that increases the ISP's ability to divide FIB storage evenly amongst different APRs. Table VI presents the near even distribution of the covered prefixes within the global routing table (from RouteViews on Dec. 1, 2008) amongst the 8 APRs.

Algorithm 1 Assignment of Virtual Prefixes to APRs

```
Select one of the eight APRs
for all VPs 0/8 to 255/8 do
  Assign VP to selected APR
  if Number of entries in APR $>$ 1/8 of GRT then
    Select a previously unselected APR
  end if
end for
```

TABLE VII
FIB SAVING FOR APR AND NON-APR ROUTERS

Routing Table break down	APR	Non-APR
Virtual Prefixes	255	256
Peer Label	20K	20K
Covered Prefixes	36999	0
Percent FIB Savings	80%	93%

Virtual prefixes are assigned to APRs using Algorithm 1. Since the global routing table changes over time, Algorithm 1 can be run occasionally from time-to-time to maintain an even distribution of FIB storage responsibilities amongst the APRs.

2) FIB Savings: FIB savings for APR and non-APR routers is calculated by counting the number of FIB entries in both APR and non-APR routers, and comparing it to the number of FIB entries that routers store in today's routing architecture. In today's architecture the FIB table contains the entire global routing table. Under VA architecture, the non-APR routers need to FIB install the virtual prefixes and peer-to-label mapping for each external router that peers with the ISP. These peer-to-label mappings facilitate the correct forwarding of encapsulated packet from APR to the appropriate egress router, which then decapsulates the packet and delivers it to the external peer. As shown in Figure 2, the FIB of non-APR router B stores label I for external router C. And the APR routers need to FIB install the peer-to-label mappings used for packet encapsulation and route for the longer prefixes covered by their assigned virtual prefixes. The number of peer-to-label mappings that each router needs to FIB install can be estimated by the number of external peers for the Tier-1 ISP, which is expected to be around 20K according to the network operator.

Table VII presents FIB savings for the APR with the largest number of entries and FIB savings for non-APR routers that store the same number of entries. The virtual prefix assignment reduces the FIB size of APR router with the largest number of FIB entries by 80% and for the non-APR routers reduces the FIB size by 93%.

3) Worst-Case Stretch: Stretch is the additional time taken by the packet to exit the ISP due to sub-optimal path introduced by the Virtual Aggregation architecture. In the worst-case, the APR for a packet can be in the opposite direction of the egress router of the ISP thereby causing the packet to be stretched by a round trip distance between the ingress PoP and the nearest major PoP containing the APR. Figure 7 illustrates such a worst-case stretch where the packet destined for 0.1.2.3 from ingress non-major PoP X needs to be sent to nearest major PoP Y (Step 1) in search of APR A which encapsulates and redirects the packet back through the PoP X to the egress router (Step 2 and 3) in PoP Z before exiting the

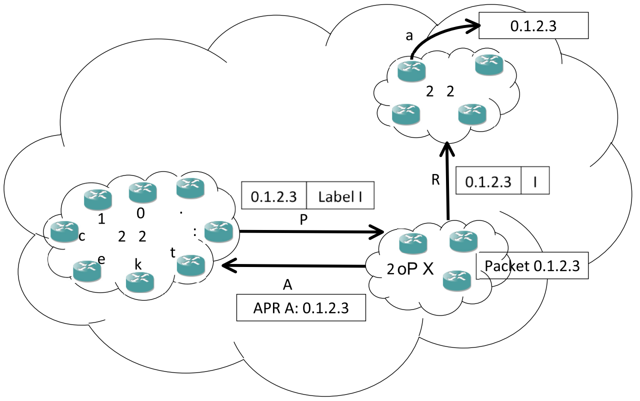


Fig. 7. Example of worst-case stretch for a packet from ingress PoP X. Worst-case stretch is the RTT distance between the ingress PoP X and the nearest PoP Y containing the appropriate APR.

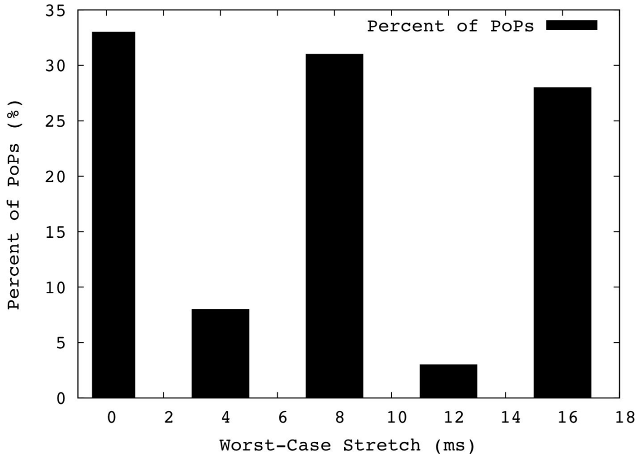


Fig. 8. Worst-case stretch evaluation results

ISP (Step 4).

We use traceroute to capture actual delay, which would include processing time and propagation time, To determine the worst case stretch for any given PoP, say PoP X, we traceroute from PoP X to all of the major PoPs containing the full set of APRs. Thereafter the worst-case stretch is the RTT distance between PoP X and its nearest major PoP where packet can be sent for an APR lookup. Figure 8 presents the distribution of the worst-case stretch for the PoP of the Tier-1 ISP. The traceroute program allowed to be used on the routers in the Tier-1 ISP rounded to the highest multiple of 4ms thereby producing 4ms gaps in-between the values of worst-case stretch times. Since all PoPs were found to be within 8ms of a major PoP the worst-case stretch experienced by packets in any PoP does not exceed 16ms. Nearly 70% of PoPs experience a worst-case stretch of 8ms or less. Moreover 32% of all PoPs experience no stretch at all because either those PoPs were major PoPs with the full set of APRs or they naturally defaulted to a major PoP for all of their traffic anyway, so VA did not change their native delivery path.

V. Related Work

The IRTF Routing Research Group [2] has been actively exploring the design space for a scalable Internet routing architecture. Most of the proposals share the same goal of resolving the scalability problem by removing the Provider-Independent (PI) prefixes and de-aggregated Provider-Allocated (PA) prefixes from the global routing system in order to facilitate the topological aggregation of prefixes in the core. We observe that all these aggregation-based proposals can be put on the same spectrum with the only difference being in the strategy of deploying aggregation in the Internet to resolve the routing scalability problem.

Several solutions follow the *elimination strategy* that attempts to enforce provider-based address aggregation throughout the Internet by eliminating all PI prefixes and de-aggregated PA prefixes. In order to facilitate such elimination each multihomed edge network takes its address assignments out of a larger aggregated block announced by each provider and thus the edge network has multiple PA addresses (one from each provider). Each end host in the multihomed site needs to be upgraded to understand how to utilize the multiple PA addresses for packet delivery. Shim6 [6] is an elimination scheme that proposes to augment the IP layer for this purpose. Shim6 defines a shim sublayer, placed in the IP layer, which ensures that the transport layer at both ends of the communication sees the same IP identifiers, even though different IP addresses can be used to forward packets along different paths.

Several other solutions follow the *separation strategy* that attempts to insert a control and management layer between edge networks and the transit core. In such scenario edge networks are no longer allowed to participate in the transit core routing nor announce their prefixes into it. A number of Map & Encap [8] schemes have been developed, including APT [14], LISP [11] and TRRP [12], which use IP-in-IP encapsulation to carry the packets across the transit core. There are also other types of separation solutions besides Map & Encap. For example, Six-One Router [25] and GSE [19] use address rewriting, which rewrites the packet header to include information about the destination's attachment point to the transit core. A common requirement of all the separation solutions is a mapping system that associates an edge prefix with the corresponding transit addresses. APT propagates the full mapping table to each AS and allows each AS to decide how to manage and store the mapping tables. On the other hand LISP has proposed a number of different mapping system designs, including LISP-CONS [7] and LISP-ALT [10]. CONS and ALT keep the mapping information at the originating edge networks, and build a global hierarchy of servers to forward mapping requests and replies. GSE proposes to store the mapping information in DNS similar to TRRP thereby avoiding the need for a new Mapping system.

In [9], Draves *et al.* designed an algorithm that aggregates a FIB to the furthest extent without introducing extra routable space. Suri *et al.* extended the ORTC work by considering each routing table entry as a 3-tuple (src, dest, action) [24]. They used dynamic programming to optimize the routing table size. In [28], we designed four levels of FIB aggregation, each level with higher aggregation ratio but also higher algorithmic complexity. By exploiting the tradeoff between extra routable space and aggregation ratio, our Level-4 algorithm can compress FIBs more than ORTC does. Moreover, previous FIB aggregation algorithms, with the exception of ours, do not handle dynamic routing updates efficiently.

VI. Conclusion

While much of the Internet community has agreed to scale routing by means of aggregation, it is still an open issue how specifically aggregation should be used to lead to a scalable Internet architecture. In this paper we show that aggregation can be adopted in an evolutionary manner, starting from a router and then within a network and then gradually expanding to include more and more networks, to address the FIB, RIB and churn scaling problems. The paper argues for an evolutionary path for moving the global routing system towards scalability with an incentive-driven adoption of aggregation within the Internet. We have used an existence proof to refute the criticism that local scope aggregation must stop at a local optimal. Furthermore, we have shown that local aggregation techniques can offer attractive tradeoffs to adopting networks without the deployment barriers inherent in other popular scalability proposals. Only time will tell whether these tradeoffs are enough to entice ISPs to adopt these steps as immediate relief to their scalability issues.

VII. Acknowledgements

This work was supported by NSF Grants 0721859, 0721369, 0721863 and 0721645.

References

[1] BGP4.net wiki. http://bgp4.net.

[2] IRTF Routing Research Group. http://www/www.irtf.org/.

[3] The RouteViews Project. http://www/routeviews/org/.

[4] H. Ballani, P. Francis, T. Cao, and J. Wang. ViAggre: Making Routers Last Longer! In *Proc. ACM Workshop on HotNets-VII*, Oct 2008.

[5] H. Ballani, P. Francis, T. Cao, and J. Wang. Making Routers Last Longer with ViAggre. In *Proc. USENIX NSDI*, Apr 2009.

[6] J. Bi, P. Hu, and L. Xie. Shim6: Reference Implementation and Optimization. In *Networking*, pages 302–313, 2008.

[7] S. Brim, N. Chiappa, D. Farinacci, V. Fuller, and D. Meyer. LISP-CONS: A Content distribution Overlay Network Service for LISP. draft-meyer-lisp-cons-04.txt, April 2008.

[8] S. Deering. The Map and Encap Scheme for Scalable IPv4 Routing with Portable Site Prefixes. Presentation, Xerox PARC, March 1996.

[9] R. P. Draves, C. King, S. Venkatachary, and B. D. Zill. Constructing Optimal IP Routing Tables. In *Proc. IEEE INFOCOM*, 1999.

[10] D. Farinacci, V. Fuller, and D. Meyer. LISP Alternative Topology (LISP-ALT). draft-fuller-lisp-alt-02.txt, April 2008.

[11] D. Farinacci, V. Fuller, and D. Oran. Locator/ID Separation Protocol (LISP). draft-farinacci-lisp-00.txt, 2007.

[12] W. Herrin. Tunneling Route Reduction Protocol (TRRP). http://bill.herrin.us/network/trrp.html.

[13] G. Houston. Growth of the BGP Table - 1994 to Present. http://bgp.potaroo.net.

[14] D. Jen, M. Meisel, D. Massey, L. Wang, B. Zhang, and L. Zhang. APT: A Practical Tunneling Architecture For Routing Scalability. Technical Report 080004, UCLA Computer Science Department, March 2008.

[15] D. Jen, M. Meisel, H. Yan, D. Massey, L. Wang, B. Zhang, and L. Zhang. Towards A New Internet Routing Architecture: Arguments for Separating Edges from Transit Core. In *Proc. ACM Workshop on HotNets-VII*.

[16] L. Li, D. Alderson, W. Willinger, and J. Doyle. A First-Principles Approach to Understanding the Internet's Router-Level Topology. In *Proc. ACM SIGCOMM*, 2004.

[17] D. Massey, L. Wang, B. Zhang, and L. Zhang. A Scalable Routing System Design for Future Internet. In *Proc. ACM Workshop on IPv6*, 2007.

[18] D. Meyer, L. Zhang, and K. Fall. Report from the IAB Workshop on Routing and Addressing. draft-iab-raws-report-01.txt, 2007.

[19] M. O'Dell. GSE:An alternate addressing architecture for IPv6. draft-ietf-ipngwg-gseaddr-00.txt, February 1997.

[20] R. Oliveira, R. Izhak-Ratzin, B. Zhang, and L. Zhang. Measurement of Highly Active Prefixes in BGP. In *Proc. IEEE GLOBECOM*, 2005.

[21] R. V. Oliveira, B. Zhang, and L. Zhang. Observing the Evolution of Internet AS Topology. *SIGCOMM Comput. Commun. Rev.*, 2007.

[22] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang. BGP routing stability of popular destinations. In *Proc. ACM Workshop on IMW'02*, 2002.

[23] P. Smith. BGP Multihoming Techniques. Nanog 23, October 2001.

[24] S. Suri, T. Sandholm, T. S, and P. Warkhede. Compressing two-dimensional routing tables. Algorithmica, 35:287-300, 2003.

[25] C. Vogt. Six/one router: A scalable and backwards compatible solution for provider-independent addressing. In *Proc. MobiArch '08*, 2008.

[26] X. Xu and P. Francis. Simple tunnel endpoint signaling in BGP. draft-xu-tunnel-00.txt, March 2009.

[27] B. Zhang and L. Zhang. Evolution Towards Global Routing Scalability. draft-zhang-evolution-01.txt, 2009.

[28] X. Zhao, Y. Liu, L. Wang, and B. Zhang. On the Aggregatability of Router Forwarding Tables. In *IEEE INFOCOM*, March 2010.